

500.39508X00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE



Applicant(s): Takashi KOBAYASHI, et al
Serial No.:
Filed: February 2, 2001
Title: METHOD FOR COOPERATING MULTIPLE
APPLICATION PROGRAMS
Group:

LETTER CLAIMING RIGHT OF PRIORITY

Honorable Commissioner of
Patents and Trademarks
Washington, D.C. 20231

February 2, 2001

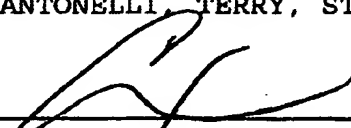
Sir:

Under the provisions of 35 USC 119 and 37 CFR 1.55, the applicant(s) hereby claim(s) the right of priority based on Japanese Patent Application No.(s) 2000-030214 filed February 2, 2000.

A certified copy of said Japanese Application is attached.

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP



Carl I. Brundidge
Registration No. 29,621

CIB/nac
Attachment
(703) 312-6600

09/773568

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
る事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
this Office.

願 年 月 日
Date of Application:

2000年 2月 2日

願 番 号
Application Number:

特願2000-030214

願 人
Applicant(s):

株式会社日立製作所

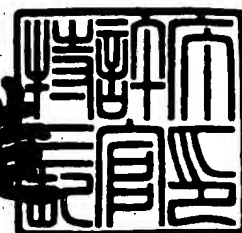
CERTIFIED COPY OF
PRIORITY DOCUMENT

Best Available Copy

2001年 1月12日

特許庁長官
Commissioner,
Patent Office

及 川 耕 造



【書類名】	特許願
【整理番号】	K00001311
【提出日】	平成12年 2月 2日
【あて先】	特許庁長官殿
【国際特許分類】	G06F 17/00
【請求項の数】	1
【発明者】	
【住所又は居所】	神奈川県川崎市幸区鹿島田 8 9 0 番地 株式会社日立製作所 ビジネスソリューション開発本部内
【氏名】	小林 隆
【発明者】	
【住所又は居所】	神奈川県川崎市幸区鹿島田 8 9 0 番地 株式会社日立製作所 ビジネスソリューション開発本部内
【氏名】	玉樹 正人
【発明者】	
【住所又は居所】	神奈川県川崎市幸区鹿島田 8 9 0 番地 株式会社日立製作所 ビジネスソリューション開発本部内
【氏名】	小川 誠
【発明者】	
【住所又は居所】	神奈川県川崎市幸区鹿島田 8 9 0 番地 株式会社日立製作所 ビジネスソリューション開発本部内
【氏名】	青山 和之
【発明者】	
【住所又は居所】	神奈川県川崎市幸区鹿島田 8 9 0 番地 株式会社日立製作所 ビジネスソリューション開発本部内
【氏名】	柏本 隆志
【発明者】	
【住所又は居所】	神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社日立製作所 ソフトウェア事業部内

【氏名】 小林 敦

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

【代理人】

【識別番号】 100075096

【弁理士】

【氏名又は名称】 作田 康夫

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 アプリケーション統合のための連携方法

【特許請求の範囲】

【請求項 1】

計算機システム群、計算機プログラム群、情報記憶手段群、ビジネスプロセス群からなるアプリケーション群を統合管理する計画管理プログラムの適用において、該計画管理プログラムにより処理を行うために必要な入出力情報群をデータモデルとして提供し、それらの情報を生成あるいは利用する業務群とそれらの実行手順をビジネスプロセスモデルとして提供し、これらのビジネスプロセスモデルとデータモデルに従って、計画管理プログラム、既存または新規のアプリケーション群を変更することなく連携させるための連携機能仕様およびそのプログラムを提供することを特徴とするアプリケーション統合のための連携方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

企業、公共機関等における業務実施のために使われる計算機システムにおけるアプリケーション連携方法に関する。

【0002】

【従来の技術】

企業、公共機関等の組織の基幹業務である会計、人事管理、生産管理、販売管理等には、合理化のための多くの投資がなされており、大抵の場合、計算機システムが適用されている。業務実施のために開発する計算機プログラム、ビジネスプロセス、データベース、ファイル等をアプリケーションとよぶ。従来のように、業務環境が安定しており製品やサービスが変わらない時代は、個々の業務部門は与えられた責務を果たしていればよかった。従って、アプリケーションは各業務に対応して個別に開発され、アプリケーション同志が情報の共有を行う必要がなかった。しかし、今日のようにマーケットニーズが目まぐるしく変化する時代になると、各業務が周辺の業務と柔軟に協調して、より大きな顧客満足を生む新しい製品やサービスを提供する必要がでてきた。

【 0 0 0 3 】

このような理由から、企業内の複数のアプリケーションを統合管理して新しい付加価値を生み出すキラーアプリケーションとよばれる計画管理プログラムパッケージが開発された。キラーアプリケーションでは、通常、既存アプリケーションが有する大量のデータを高速処理するとともに、その結果を広範囲で共有する必要がある。そこで、高性能なデバイス、大容量のメモリー／ディスク、高速アルゴリズム、インターネット等の最新情報技術を活用したものが多い。例えば、製造分野における高速所要量計算プログラム、販売分野における販売データ多次元分析プログラム等である。

【 0 0 0 4 】

キラーアプリケーションを業務システムに適用する作業は、通常、次の２段階の手順で行う。

【 0 0 0 5 】

〔ステップ１〕業務設計：キラーアプリケーションを前提として、ビジネスプロセスや情報を共有し全体最適化を実現する新業務手順を設計する。

【 0 0 0 6 】

〔ステップ２〕システム設計・実装：キラーアプリケーション、既存アプリケーション間でビジネスプロセスや情報を共有するための方式設計とソフトウェア開発を行う。

【 0 0 0 7 】

このように、キラーアプリケーションの適用は個別業務に閉じたものでなくビジネスプロセス全体に関わるため、大抵の場合、その作業工数は膨大なものとなる。そこで、従来から工数低減の方法が検討されてきた。

【 0 0 0 8 】

上記の２つの手順のうち、システム設計・実装に関しては早くから問題点が分析され改善策が考案されている。キラーアプリケーションと既存アプリケーションとの間の情報共有の問題点は以下の通りである。すなわち、個々のアプリケーションは各業務に対応してバラバラに開発されているため、アプリケーション間で情報を受渡すための仕掛けは具備していない。従って、キラーアプリケーション

ンを適用するためには、周辺システムとのインタフェース処理プログラムを新たに開発しなければならない。通常、アプリケーションごとにその稼働環境、プログラミング言語、運用方法等が異なるため、連携するアプリケーションごとにインタフェース処理プログラムが必要となり、開発工数や保守工数が膨大なものとなる。この問題を解決する技術として、David S. Linthicumの「等身大で実現するEAI」（DB Magazine, 2000 January）にもあるように、アプリケーションを接続し、データを共有するエンタープライズアプリケーションインテグレーション（EAI）が脚光を浴びている。EAIでは、通常、メッセージブローカとよばれるミドルウェアを使ってアプリケーションを連携する。メッセージブローカは、複数のアプリケーション間でメッセージ形式で情報の仲介を行うものである。通常、次の機能を有する。

【0009】

メッセージ解釈：アプリケーションに合わせてメッセージの表現形式を変換する。

【0010】

知的ルーティング：メッセージを識別して適切なアプリケーションに発送する。

【0011】

ルールエンジン：メッセージ解釈とルーティングのルールを処理する。

【0012】

上記の機能を利用することにより、インタフェース処理プログラムの開発工数を大幅に削減できるようになった。

【0013】

【発明が解決しようとする課題】

以上に述べたように、EAI技術により、キラーアプリケーション適用時のシステム設計・実装に関わる問題は改善された。それに伴い、それに先行する「全体最適化を実現する新業務手順の設計」という業務設計に関わる問題がクローズアップされてきた。従来、業務設計の工数低減に対する解決策は、キラーアプリケーションのベンダー、SIベンダー、コンサルティングファーム等により、

業務テンプレートという形態で提供されている。業務テンプレートとは、キラーアプリケーションの効力を発揮するための推奨するビジネスプロセスモデルとデータモデルを示したものであり、レファレンスモデルともよばれる。業務テンプレートを手本とすることにより、ゼロベースで行うよりも低工数で業務設計を行うことを狙っている。しかし、現実には、ユーザごとに業務の目的や制約条件が異なるため、業務テンプレートと実業務との間で妥協点を見出すアプローチをとる。すなわち、業務テンプレートに合わせて実業務を変更する、あるいは、実業務に合わせて業務テンプレートをカスタマイズする。この調整作業には、次の人材による知識集約的な検討が必要となる。

【0014】

業務エキスパート：実業務を熟知しかつ必要に応じて業務スタイルを変更する権限をもつ。通常、業務ごとに分散している。

【0015】

プロジェクト責任者：実業務の変更、あるいは、業務テンプレートのカスタマイズの最終的な決定権限をもつ。

【0016】

コンサルタント：キラーアプリケーションと業務テンプレートを熟知し、カスタマイズによる費用と効果を評価ができる。

【0017】

通常、このような優秀な人材を見出すことは難しく、たとえ存在しても極めて多忙なことが多い。従って、この調整作業がキラーアプリケーション適用のボトルネックとなり、プロジェクト期間が長期化してしまう。

【0018】

本発明の目的は、以上に述べた問題を解決し、キラーアプリケーション適用時の業務設計を低工数かつ短時間で実現することにある。

【0019】

【課題を解決するための手段】

業務テンプレートと実業務の調整に時間を要するのは、業務テンプレートが個別性の高い詳細仕様まで規定しているからである。その結果、業務テンプレート

に従ってシステムを構築した場合、業務の運用方法を全面的に変更する必要がある。さらに、職制、社内規則、取引先との契約等の変更にもつながる。

【 0 0 2 0 】

そこで、本発明の業務テンプレートでは、統合の対象となる個別の業務内の手順は規定せず、業務間の手順だけを規定する。そもそもキラーアプリケーションは特定のユーザ向けに開発されたものではなく、想定されたドメインに対して汎用的に作られている。従って、キラーアプリケーションが前提としているビジネスプロセスモデルとデータモデルに従って、統合すべき業務間のマクロな手順を規定する。ただし、この業務テンプレートには個別性の高い詳細部分が規定されていないため、それだけで実業務を行うことは困難である。そこで、業務内の詳細部分に、現行ビジネスプロセスや既存アプリケーションをそのまま変更することなく組込む。そのために、次のシステム連携機能を設ける。

【 0 0 2 1 】

業務実行管理：業務テンプレートに規定されたビジネスプロセスモデルに従って進行状況を管理し、実行すべき業務を認識し指示する。

【 0 0 2 2 】

業務開始指示：業務実行管理の指示により、当該業務を実行するためのアプリケーション群や人間に対して実行開始の信号を送出する。

【 0 0 2 3 】

業務完了検知：アプリケーション群や人間が業務を実行完了したことを示す信号を検知する。

【 0 0 2 4 】

業務結果入手：業務の実行結果を示す情報を入手し、キラーアプリケーションに渡す。必要に応じて、情報構造、形式、値の変換を行う。

【 0 0 2 5 】

計画情報通知：キラーアプリケーションの処理結果を、アプリケーション群や人間に対して通知する。必要に応じて、情報構造、形式、値の変換を行う。

【 0 0 2 6 】

以上に述べたマクロレベルの業務テンプレートとシステム連携機能により、現

行の業務スタイルの変更や業務テンプレートのカスタマイズを行うことなく、低工数かつ短期間でキラーアプリケーションを適用できる。

【0027】

ただし、当然のことであるが、個別業務内に組込んだ現行業務部分が、キラーアプリケーション適用後の新業務スタイルに適合している保証はない。しかし、膨大な工数と時間をかけて全体最適化を狙うよりは、多少の不具合があっても低工数、短期間でキラーアプリケーションを適用する方がリスクが少なくビジネス上の効果も大きい。個別業務内の改善は、キラーアプリケーションを適用しながら段階的に行うべきである。

【0028】

以上の議論を踏まえて、本発明の業務テンプレートの内容、連携機能の仕様、および、キラーアプリケーションの適用方法を詳しく説明する。

【0029】

(1) 業務テンプレート内容

キラーアプリケーションによる計画管理処理を実行するために必要なビジネスモデルとデータモデルを次のように規定する。

【0030】

ビジネスプロセスモデル：キラーアプリケーションによる計画管理処理を行うために必要な業務群とそれらの実行手順を示す。通常、キラーアプリケーションでは、その計画管理処理に必要な入出力情報群のみを規定する。従って、それらの情報を生成する、あるいは、利用する業務群がビジネスプロセスモデルの対象となる。このビジネスプロセスモデルは、下記に示す連携機能の業務実行管理（ワークフロー管理）を行うためのビジネスプロセス定義となる。

【0031】

データモデル：キラーアプリケーションの入出力情報の実体（エンティティ）群とそれらの関連、構造、形式を示す。このデータモデルはキラーアプリケーションの入出力ファイルの定義となる。また、下記に示す連携機能の業務結果入手と計画結果通知において、情報変換ルールを定義するために利用する。

【0032】

(2) 連携機能仕様

上記の業務テンプレートに対して、既存のアプリケーションとビジネスプロセスを変更することなく連携させるために、次の連携機能を設ける。

【0033】

業務実行管理：いわゆるワークフロー管理であり、業務テンプレートに規定されたビジネスプロセスモデルに従って進行状況を管理し、実行すべき業務を認識し指示する。

【0034】

業務開始指示：業務実行管理の指示により、当該業務を実行するためのアプリケーション群や人間に対して実行開始の信号を送出する。例えば、対象が計算機プログラムやビジネスプロセスならば起動コマンド、人間ならば開始通知メールを送出する。

【0035】

業務完了検知：アプリケーション群や人間が業務を実行完了したことを示す信号を検知する。例えば、計算機プログラムやビジネスプロセスの終了コマンドの傍受、データベースやファイル等の情報記憶手段における値の変更監視、メールの受付による。

【0036】

業務結果入手：業務の実行結果を示す情報を入手し、キラーアプリケーションに渡す。この処理の実行タイミングは、上記の業務完了検知により判断する。必要に応じて、情報構造、形式、値の変換を行う。情報変換ルールは、上記で説明したキラーアプリケーションの入力情報のデータモデルと業務の実行結果情報のデータモデルとの対応関係により定義する。

【0037】

計画情報通知：キラーアプリケーションの処理結果を、アプリケーション群や人間に対して通知する。必要に応じて、情報構造、形式、値の変換を行う。この情報変換ルールは、キラーアプリケーションの出力情報のデータモデルと業務の計画情報のデータモデルとの対応関係により定義する。

【0038】

(3) キラーアプリケーション適用方法

上記の業務テンプレートと連携機能に基づき、次の手順に従って、キラーアプリケーションを業務システムに適用する。

【0039】

[ステップ1] 業務設計

要件分析：新業務として実現する機能を決定する。この際、利用するキラーアプリケーションを選定し、その機能を最大限に活用することを検討する。

【0040】

ビジネスプロセス設計：業務テンプレートのビジネスプロセスモデルの各ステップに、キラーアプリケーション、あるいは、実業務におけるアプリケーション群を割り当て、新業務が実現できることを確認する。

【0041】

データモデル設計：業務テンプレートのデータモデルの各実体に対して、実業務におけるファイル、データベース、メッセージ等の情報記憶手段を割り当て、必要情報が存在することを確認する。

【0042】

業務テンプレートによって要求されたアプリケーション群が存在しない場合は、必要に応じて新規に設計する。

【0043】

[ステップ2] システム設計

入出力設計：キラーアプリケーションと、実業務におけるアプリケーション群との間の入出力フローを設計する。

【0044】

連携方式設計：業務テンプレートのビジネスプロセスモデルに従って、アプリケーション群を起動したり、終了検知するための連携機能の仕様を決定する。

【0045】

データ変換方式設計：キラーアプリケーションの入出力ファイル（業務テンプレートのデータモデルとして記述）と実業務の情報記憶手段との間の変換仕様を

設計する。

【0046】

[ステップ3] 実装

連携部品開発：上記連携機能の仕様に従ったプログラムを開発する。

【0047】

システム実装：業務テンプレートのビジネスプロセスモデルをワークフローツールで実装し、データモデルをデータベースツールで実装する。さらに、キラーアプリケーションと、実業務におけるアプリケーション群を連携部品を利用して連携する。

【0048】

[ステップ4] 拡張

段階的拡張：ビジネスプロセスモデルの各業務ステップに連携した実業務部分を、キラーアプリケーションを適用しながら段階的に拡張する。この拡張は、アプリケーションパッケージの適用、新規ユーザプログラムの開発、ワークフローの導入等による。

【0049】

【発明の実施の形態】

以下、本発明の実施例を図1から図10に従って示す。

【0050】

図1は本発明のアプリケーション統合のための連携基盤の全体構成を示すものである。図において100はアプリケーション連携基盤の本体であり、業務実行機能(101)、開発支援機能(102)、業務開始指示機能(103)、業務完了検知機能(104)、業務結果入手機能(105)、計画情報通知機能(106)の6機能により構成する。また、これらの機能群によりアプリケーション統合を行うために、開始完了ルール(107)、情報変換ルール(108)、ビジネスプロセス定義(109)、業務テンプレート(110)の4つの情報記憶手段を用いる。一方、図において120は、100のアプリケーション連携基盤の対象となるアプリケーション群であり、様々な計算機プログラム(121)、ワークフローのビジネスプロセス(122)、データベース(123)、人間(

124) 等により構成する。

【0051】

101の業務実行管理はいわゆるワークフロー管理機能であり、110の業務テンプレートに規定されたビジネスプロセスモデルに従って進行状況を管理し、実行すべき業務を認識し指示する。103の業務開始指示は、101の業務実行管理の指示により、当該業務を実行するためのアプリケーション群や人間に対して実行開始の信号を送出する。例えば、121の計算機プログラムや122のビジネスプロセスに対しては起動コマンドを送出し、123のデータベースに対しては予め定めたデータ項目に開始指示信号を書込み、124の人間ならば開始通知メールを送出する。104の業務完了検知は、アプリケーション群や人間が業務を実行完了したことを示す信号を検知する。例えば、121の計算機プログラムや122のビジネスプロセスの終了コマンドを傍受し、123のデータベースにおける予め定めたデータ項目値の変更監視を行い、124の人間から業務完了メールを受付ける。このように、アプリケーションの種類に応じて異なる開始指示仕様と完了検知仕様は、開発支援102により予め定義し開始完了ルールとして107に記憶しておく。105の業務結果入手は、120のアプリケーション群における計算機プログラム、データベース等から業務の実行結果を示す情報を入手し、120におけるキラーアプリケーションに渡す。この処理の実行タイミングは、104の業務完了検知により判断する。106の計画情報通知は、120のアプリケーション群におけるキラーアプリケーションの処理結果を、120における計算機プログラム、データベース等に対して通知する。これらの業務結果入手と計画情報通知においては、必要に応じて、情報構造、形式、値の変換を行う。その際の情報変換仕様は、開発支援102により予め定義し情報変換ルールとして108に記憶しておく。110の業務テンプレートは、キラーアプリケーションによる計画実行処理を実行するために必要なビジネスプロセスモデルとデータモデルを示すものであり、102の開発支援を用いて実業務に応じてカスタマイズできる。109のビジネスプロセス定義は、101の業務実行管理を行うためのワークフローツールの定義情報であり、110の業務テンプレートにおけるビジネスプロセスモデルに基づいて、102の開発支援を用いて定義する。

【 0 0 5 2 】

以下、本発明の実施例を、組立て製造業における部品調達業務を題材にして説明する。本実施例では、キラーアプリケーションとして高速所要量計算プログラムを採用する。これにより、既存の生産管理業務と部品発注業務との間で生産計画情報、部品在庫情報、部品メーカーへの注文情報等を共有し、従来に比べて在庫を削減し、ビジネスリードタイムを短縮する。

【 0 0 5 3 】

まず初めに、本発明の業務テンプレートについて説明する。図 2 は業務テンプレートにおけるビジネスプロセスモデルであり、キラーアプリケーションによる計画管理業務処理を行うために必要な業務群とそれらの実行手順だけを示す。以下に、ビジネスプロセスモデルの規定方法を述べる。本実施例のキラーアプリケーションは高速所要量展開プログラムであり、発注管理と生産計画に適用される。そして、注残、在庫、生産計画の 3 つの情報を入力し、新たに依頼すべき部品量を出力する。注残情報と在庫情報を生成するのは発注管理業務であり、生産計画情報を生成するのは生産計画業務である。また、これら 2 つの業務は高速所要量展開の結果である依頼部品情報を利用する。従って、ビジネスプロセスモデルとしては、発注管理業務と生産計画業務だけを対象とし、それらと高速所要量展開プログラムとの間の情報授受を行うための最小限の実行順序関係を示せばよい。

【 0 0 5 4 】

図 2 にその結果であるビジネスプロセスモデルを示す。図において、200 は発注管理のビジネスプロセスモデルであり、必要部品をタイムリーに調達する発注管理業務（201）と、部品手配量をリアルタイムに計算する所要量展開業務（205）により構成する。201 の発注管理業務は、手配依頼作成（202）、発注（203）、入荷・検収（204）の 3 つのサブ業務により構成する。202 の手配依頼作成業務では、高速所要量展開プログラムを適用して日々のリアルタイムな情報に基づいて部品手配量を算出し依頼部品データベース 209 に登録する。203 の発注業務では、209 の依頼部品データベースの情報に基づいて各取引先に対して発注伝票を通知するとともに、その結果を注残データベース

210に登録する。204の入荷・検収業務では、発注部品が取引先から到着するとそれを検収し、注残データベースと在庫データベースに登録する。205は所要量展開業務であり、データ入手(206)、所要量展開(207)、依頼部品抽出(208)の3つの処理により構成する。この所要量展開業務では、206のデータ入手処理において生産計画(230)、注残(210)、在庫(211)の3つの情報を入力し、207の所要量展開処理を実施した後に、208の依頼部品抽出処理において依頼部品情報(209)を出力する。

【0055】

一方、220は生産計画のビジネスプロセスモデルであり、製品需要と供給能力のバランスをとった生産計画をタイムリーに作成する生産計画業務(221)と、製品需要に基づく必要部品量をリアルタイムにシミュレーションする所要量展開業務(225)により構成する。221の生産計画業務は、需要予測(222)、資材確認(223)、生産計画作成(224)の3つのサブ業務により構成する。222の需要予測業務では、過去の売上実績や市場の動向等に基づいて製品需要量を予測し、生産計画の草案を作った後に生産計画データベース(230)に登録する。223の資材確認業務では、高速所要量展開プログラムを適用して、生産計画の草案に基づいて依頼部品量を算出し229の依頼部品データベースに登録する。この際、各取引先への依頼量、納期等を種々に調整し、部品の需要量と供給量をバランスさせる。224の生産計画作成業務では、最終的な依頼部品情報に基づいて、生産計画の確定案を作成し230の生産計画データベースに登録する。225は所要量展開業務であり、205と同様に、データ入手(226)、所要量展開(227)、依頼部品抽出(228)の3つの処理により構成する。

【0056】

図3は業務テンプレートにおけるデータモデルであり、キラーアプリケーションによる計画管理業務処理を行うために必要な入出力情報の実体群とそれらの関連、構想、形式を示す。すなわち、まず、生産計画(300)、注残(305)、在庫(306)、依頼部品(303)というキラーアプリケーションの3つの入力情報と1つの出力情報が、データモデルの実体を構成する。そして、それら

の情報間で重複している製品（301）、部品（302）、取引先（304）という実体をくくり出し新たに実体として定義する。320から325はそれらの実体間の関連を示す。また、各実体の情報項目の主なものを楕円で示す。

【0057】

次に、本発明の業務連携機能について説明する。図4は業務連携機能における業務開始指示機能と業務完了検知機能で用いる開始完了ルールを示す。この表において、401の列は図2に示す各ビジネスプロセスを構成する業務名称を示す。402の列は各業務で実行するアプリケーション名称を示し、403の列は当該アプリケーションを開始するためのルール、404の列は当該アプリケーションの完了を検知するためのルールを示す。

【0058】

図5は業務開始指示機能の処理手順を示し、以下の通りである。

【0059】

手順500：業務実行管理機能から開始指示信号SRART(業務名称)を受取る。
この信号は、ビジネスプロセス定義において新たな業務ステップに変わった時に、業務実行管理機能から業務開始指示機能に対して送出される。

【0060】

手順501：図4に示す開始完了ルールにおいて、当該業務の開始指示ルールを調べる。

【0061】

手順502：当該開始指示ルールに従って、当該業務に対して開始指示信号を送出する。

【0062】

図6は業務完了検知機能のうち、特に、データベースやファイル等の情報記憶手段における値の変更監視に基づく処理手順を示す。この手順は以下の通りである。

【0063】

手順600：本機能によるアプリケーションの完了検知を開始したタイミングで、本機能が有するタイマーに、監視のサイクルタイムTを初期値としてセット

する ($t = T$)。

【0064】

手順601: タイマー値 t が0以下か否かをチェックする ($t \leq 0$)。0以下ならば手順602に進む。そうでないならば手順604に進む。

【0065】

手順602: 指定された記憶領域の値が完了条件を満たすかチェックする。満たすならば手順603に進む。そうでないならば手順600に進む。

【0066】

手順603: 業務実行管理機能に対して当該業務の完了を報告し、手順600に進む。

【0067】

手順604: 1単位時間休止した後にタイマー値 t を1つ減じて ($t = t - 1$)、手順601に進む。

【0068】

図7は業務連携機能における業務結果入手機能と計画情報通知機能で用いる情報変換ルールを示す。本実施例では、業務結果入力機能において、実業務における発注管理データベース(700)と検収管理データベース(710)を、高速所要量展開プログラムにおける注残データベース(720)に情報変換する場合のルールを示す。720の注残データベースの情報項目である721から725のうち、724の発注残項目を除く全ては、700の発注管理データベースの情報項目と1対1に対応する。従って、これらの情報項目については、データ型と桁数の変換だけを行えばよい。一方、724の発注残項目は、700における注文数項目(704)の値から、710の入荷数項目(714)の値を減じて求める。

【0069】

図8は業務結果入手機能の処理プログラムの作成方法を示す。それは以下の通りである。

【0070】

手順800: 当該業務結果入手に必要な、図7に示す情報変換ルールを入力す

る。

【 0 0 7 1 】

手順 8 0 1 : 入力ファイルの情報項目のうち、出力ファイルの情報項目の作成に必要なサブセットを抽出する。図 7 の例では、7 0 0 の発注管理データベースの 7 0 1 から 7 0 5 までの 5 つの情報項目と、7 1 0 の発注管理データベースの 7 1 4 の情報項目である。

【 0 0 7 2 】

手順 8 0 2 : 入力ファイル情報項目のサブセットについて、データベースアクセスのためのコマンド、アプリケーションインタフェースプログラム等を用いて情報読込み部を作成する。

【 0 0 7 3 】

手順 8 0 3 : 出力ファイル情報項目について、高速所要量展開プログラムのアプリケーションインタフェースプログラムを用いて情報書出し部を作成する。

【 0 0 7 4 】

手順 8 0 4 : 手順 8 0 0 で入力した情報変換ルールに基づき、入力ファイル情報項目サブセットから出力ファイル情報項目への情報変換部を作成する。

【 0 0 7 5 】

手順 8 0 5 : 上記の読込み部、書出し部、情報変換部を結合し、業務結果入手プログラムを完成させてファイルに出力する。

【 0 0 7 6 】

図 9 は計画情報通知機能の処理プログラムの作成方法を示す。それは以下の通りである。

【 0 0 7 7 】

手順 9 0 0 : 当該計画情報通知に必要な、図 7 に示す情報変換ルールを入力する。

【 0 0 7 8 】

手順 9 0 1 : 出力ファイルの情報項目のうち、入力ファイルの情報項目の作成に必要なサブセットを抽出する。

【0079】

手順902：入力ファイル情報項目について、高速所要量展開プログラムのアプリケーションインタフェースプログラムを用いて情報読み込み部を作成する。

【0080】

手順903：出力ファイル情報項目のサブセットについて、データベースアクセスのためのコマンド、アプリケーションインタフェースプログラム等を用いて情報書き出し部を作成する。

【0081】

手順904：手順900で入力した情報変換ルールに基づき、入力ファイル情報項目から出力ファイル情報項目サブセットへの情報変換部を作成する。

【0082】

手順905：上記の読み込み部、書き出し部、情報変換部を結合し、計画情報通知プログラムを完成させてファイルに出力する。

【0083】

最後に、以上に述べた業務テンプレートと業務連携機能により構成するアプリケーション連携基盤を用いて、キラーアプリケーションを既存／新規アプリケーション群と連携させる手順を、図10に示す。本手順は大きく業務設計（1000）、システム設計（1010）、実装（1020）、拡張（1030）の4つにより構成する。

【0084】

（1）業務設計

手順1001：要件分析：新業務として実現する機能を決定する。この際、適用しようとしているキラーアプリケーションの業務テンプレートにおけるビジネスプロセスモデルとデータモデルを参考にして、キラーアプリケーションの機能を最大限に活用することを検討する。

【0085】

手順1002：ビジネスプロセス設計：業務テンプレートの、図2に示すビジネスプロセスモデルの各ステップに、キラーアプリケーション、あるいは、実業務におけるアプリケーション群を割り当て、新業務が実現できることを確認する

【 0 0 8 6 】

手順 1 0 0 3 : データモデル設計 : 業務テンプレートの、図 3 に示すデータモデルの各実体に対して、実業務におけるファイル、データベース、メッセージ等の情報記憶手段を割り当て、必要情報が存在することを確認する。

【 0 0 8 7 】

(2) システム設計

手順 1 0 1 1 : 入出力設計 : キラーアプリケーションと、実業務におけるアプリケーション群との間の入出力フローを設計する。

【 0 0 8 8 】

手順 1 0 1 2 : 連携方式設計 : 業務テンプレートのビジネスプロセスモデルに従って、アプリケーション群を開始指示したり、完了検知するための連携機能の仕様を、図 4 に示すような形式で設計する。

【 0 0 8 9 】

手順 1 0 1 3 : データ変換方式設計 : キラーアプリケーションの入出力ファイル (業務テンプレートのデータモデルとして記述) と実業務の情報記憶手段との間の変換仕様を、図 7 に示すような形式で設計する。

【 0 0 9 0 】

(3) 実装

手順 1 0 2 1 : 連携部品開発 : 上記連携機能の仕様に従った業務開始指示、業務完了検知、業務結果入手、計画情報通知の各プログラムを、それぞれ、図 5、6、8、9 に示す各手順により開発する。

【 0 0 9 1 】

手順 1 0 2 2 : システム実装 : 業務テンプレートのビジネスプロセスモデルをワークフローツールで実装し、データモデルをデータベースツールで実装する。さらに、キラーアプリケーションと、実業務におけるアプリケーション群を連携部品を利用して連携する。

【 0 0 9 2 】

(4) 拡張

手順 1 0 3 1 : 段階的拡張 : ビジネスプロセスモデルの各業務ステップに連携した実業務部分を、キラーアプリケーションを適用しながら段階的に拡張する。この拡張は、アプリケーションパッケージの適用、新規ユーザプログラムの開発、ワークフローの導入等による。

【 0 0 9 3 】

【発明の効果】

本発明のアプリケーション統合のための連携方法および基盤によれば、業務テンプレートとして提供するビジネスプロセスモデルとデータモデルに従って、統合すべき業務間のマクロな手順を開発できる。さらに、業務連携機能として提供する業務実行管理、業務開始指示、業務完了検知、業務結果入手、計画情報通知の各機能を用いて、現行ビジネスプロセスや既存アプリケーションをそのまま変更することなく組込んで各業務内のミクロな手順を実現できる。このため、キラーアプリケーション適用時のシステム開発を低工数かつ短時間で実現することができる。

【 0 0 9 4 】

以上により、下記のような効果が期待できる。

【 0 0 9 5 】

顧客の効果 : 新業務を実現するシステムを短納期、低工数、高信頼で構築できる。そして、キラーアプリケーションを適用しながら、段階的にシステムの拡張を行うことができる。

【 0 0 9 6 】

システムインテグレーションベンダーの効果 : キラーアプリケーションの対象とする業務に関する詳細な知識がなくても、本発明の基盤と方法によれば、システムインテグレーションビジネスに参入できる。

【 0 0 9 7 】

ソフトウェアベンダー : 業務連携プログラムの仕様を標準化することにより、ソフトウェア部品や設計支援ツールのビジネスに参入できる。

【図面の簡単な説明】

【図 1】

本発明のアプリケーション統合のための連携基盤の全体構成図を示す。

【図 2】

本発明の業務テンプレートのビジネスプロセスモデルを示す。

【図 3】

本発明の業務テンプレートのデータモデルを示す。

【図 4】

本発明の業務開始指示機能と業務完了検知機能で用いる開始完了ルールを示す。

【図 5】

業務開始指示機能の処理手順を示す。

【図 6】

業務完了検知機能の処理手順を示す。

【図 7】

本発明の業務結果入手機能と計画情報通知機能で用いる除法変換ルールを示す。

【図 8】

業務結果入手機能の処理プログラムの作成方法を示す。

【図 9】

計画情報通知機能の処理プログラムの作成方法を示す。

【図 10】

本発明のアプリケーション統合のための連携基盤を用いて、キラーアプリケーションを既存／新規アプリケーション群と連携させるための手順を示す。

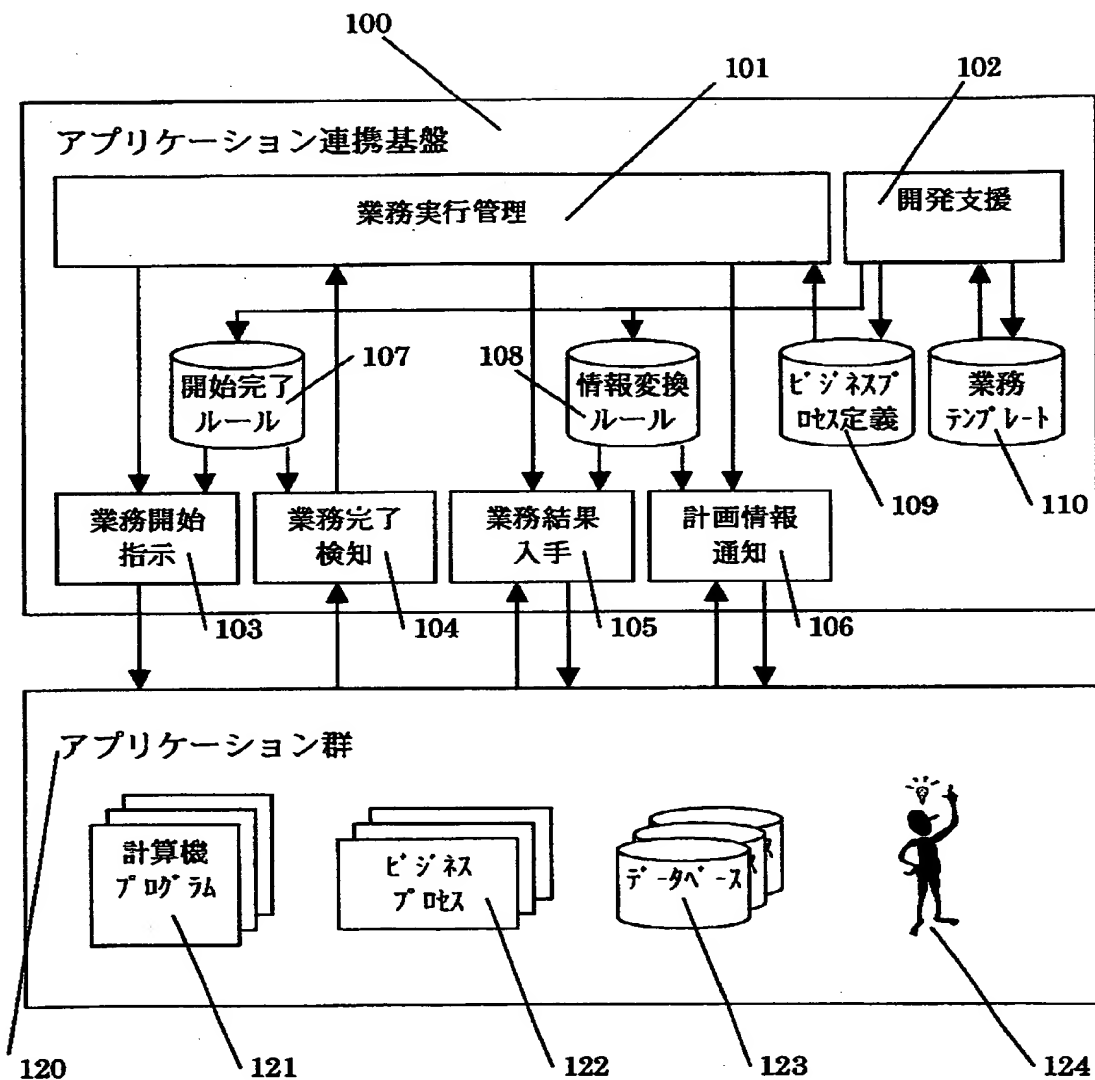
【符号の説明】

100：アプリケーション連携基盤、120：アプリケーション群、121：計算機プログラム。

【書類名】 図面

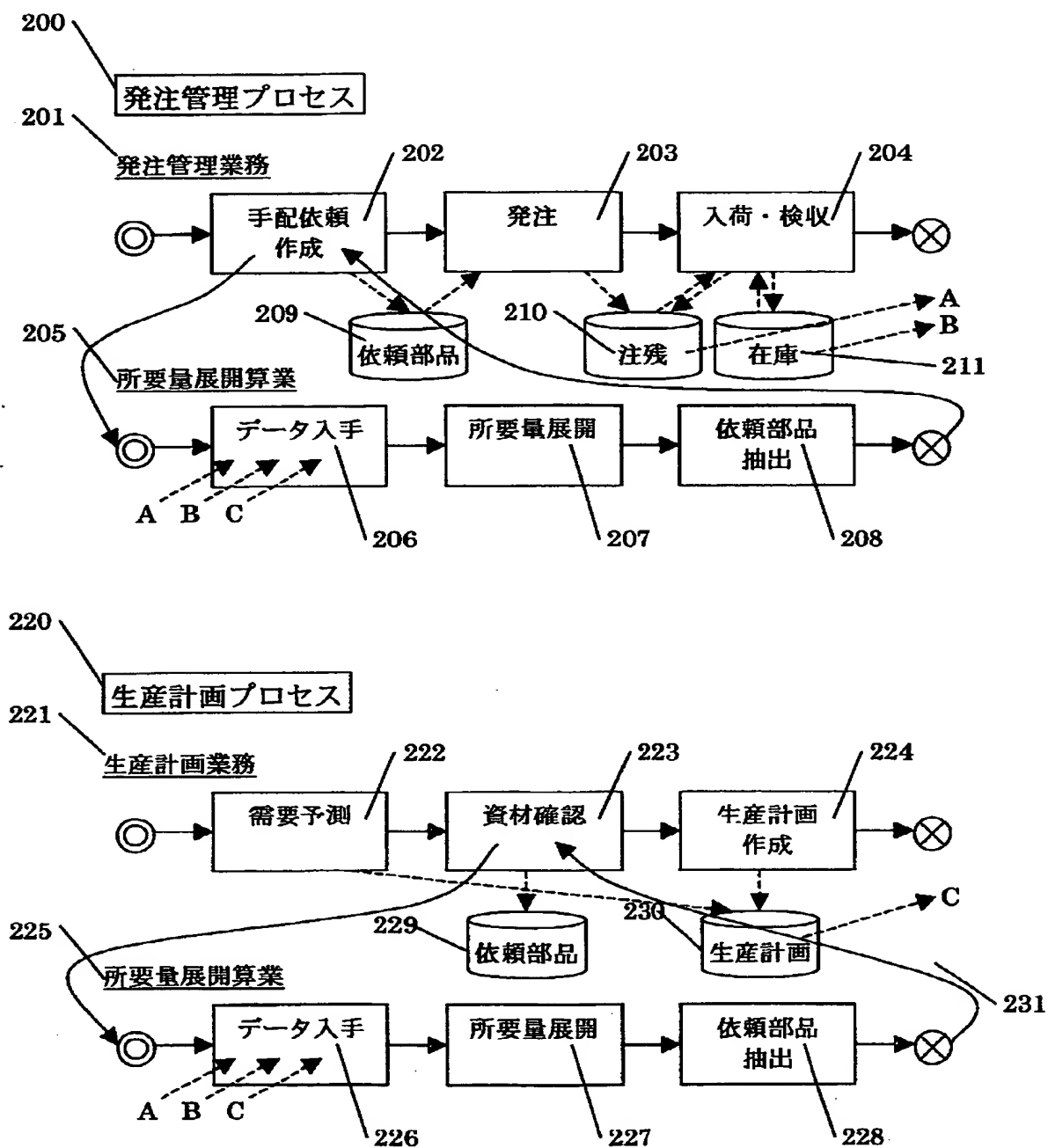
【図1】

図1



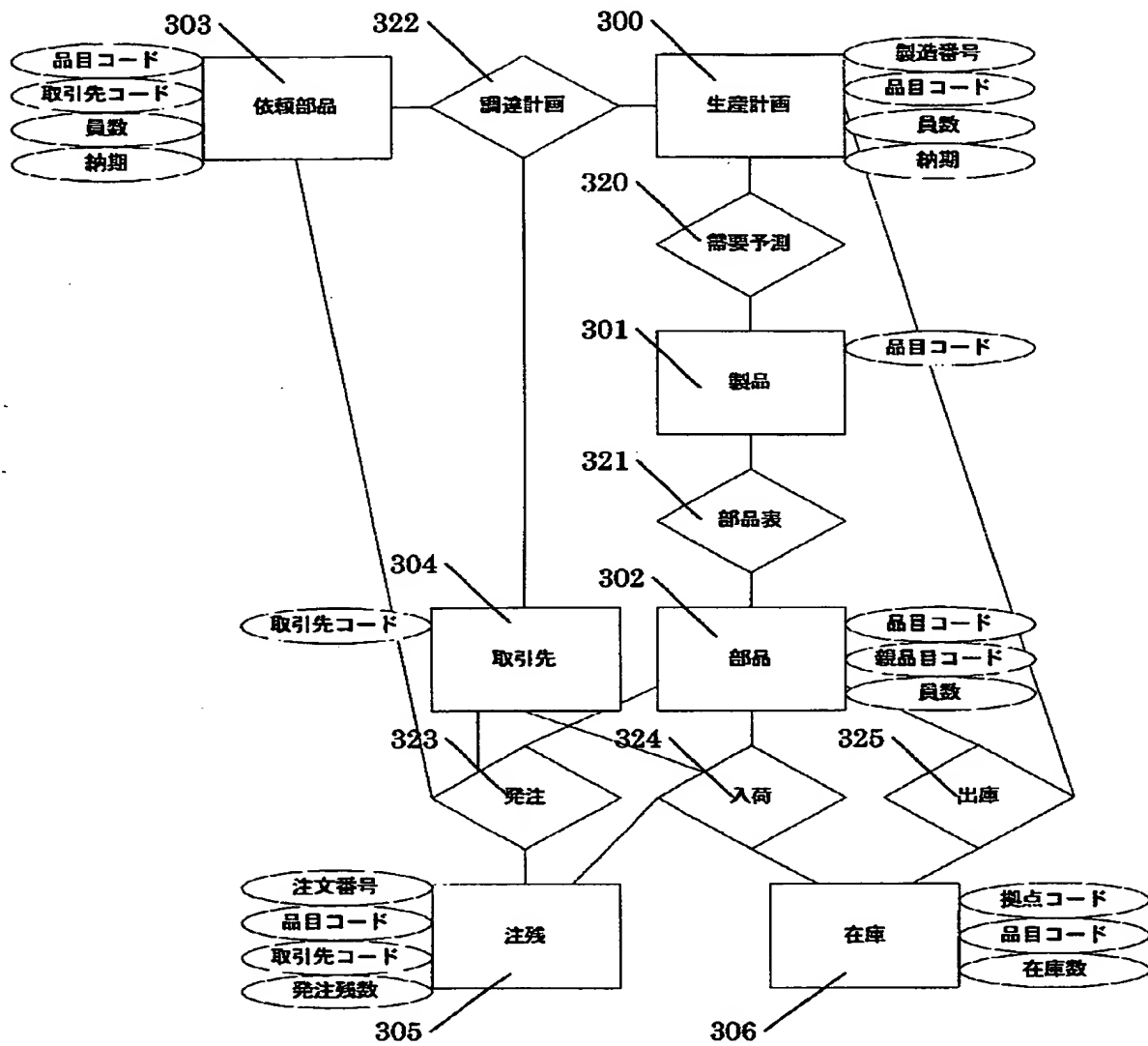
【図 2】

図 2



【図 3】

图 3



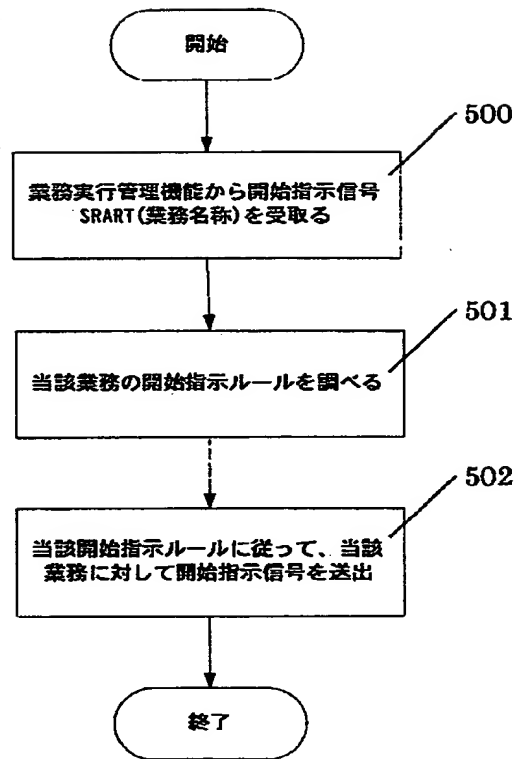
【図 4】

図 4

#	業務名称	連携アプリケーション名称	業務開始指示ルール	業務完了検知ルール
1	手配依頼作成	所要量展開BP	BP起動	BP完了信号受信
2	発注	発注管理PG	PG起動	PG完了信号受信
3	入荷・検収	検収管理PG	PG起動	検収完了フラグのON

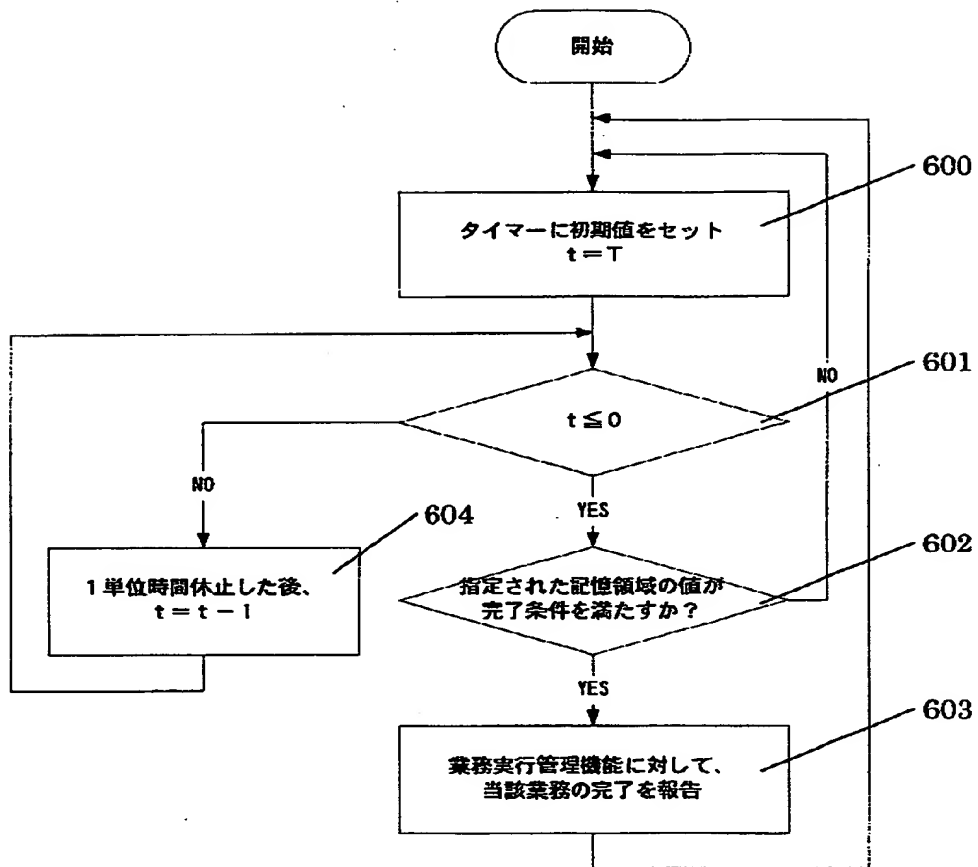
【図 5】

図 5



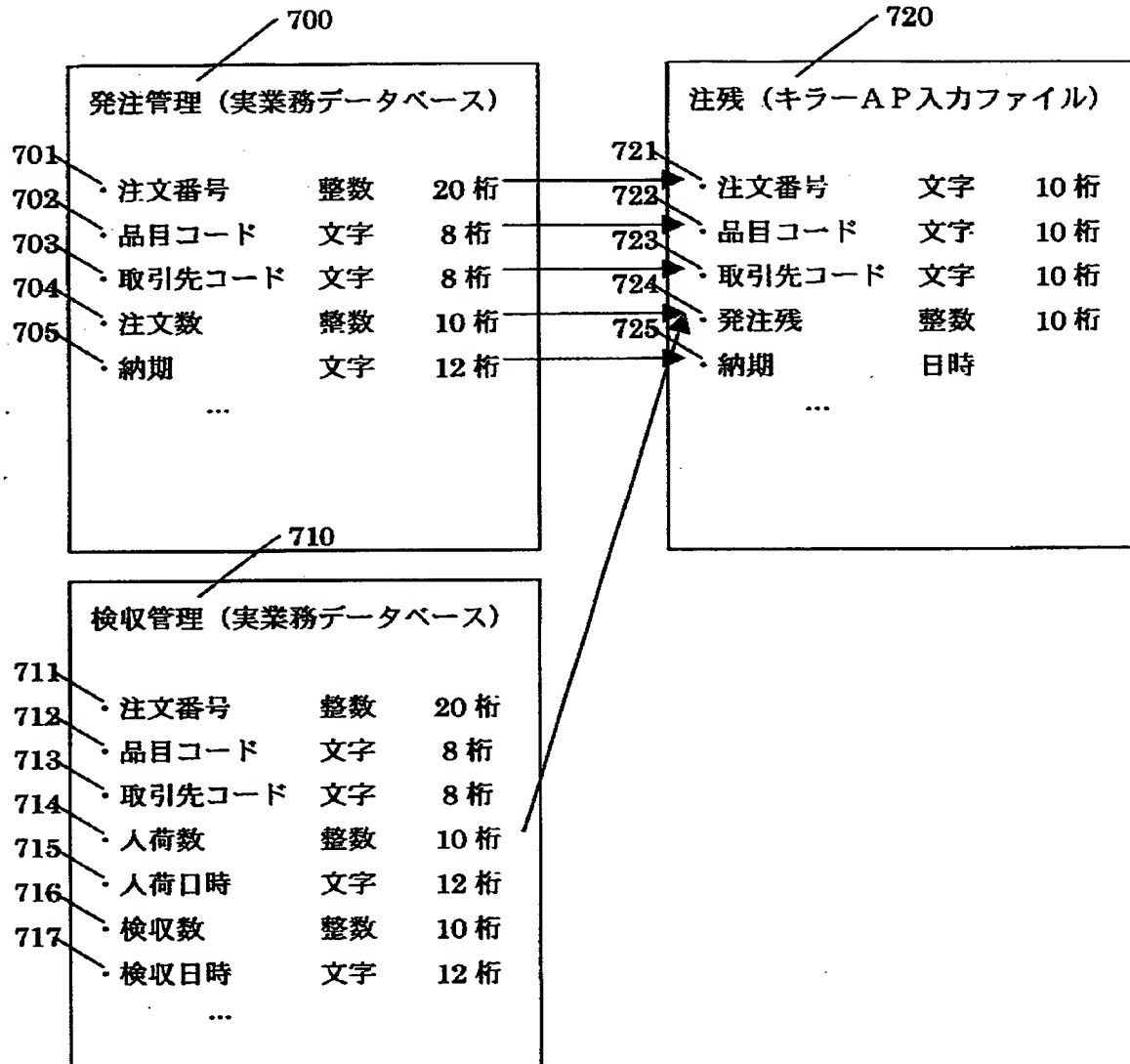
【図 6】

図 6



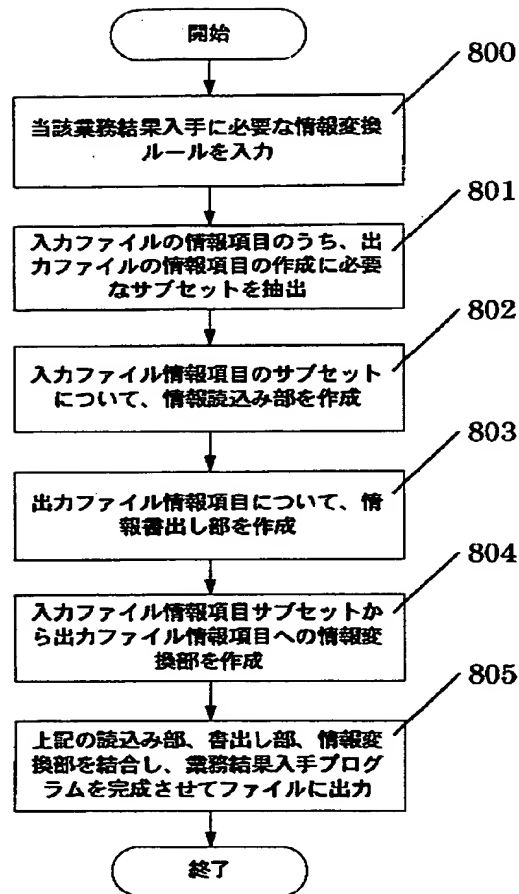
【図 7】

図 7



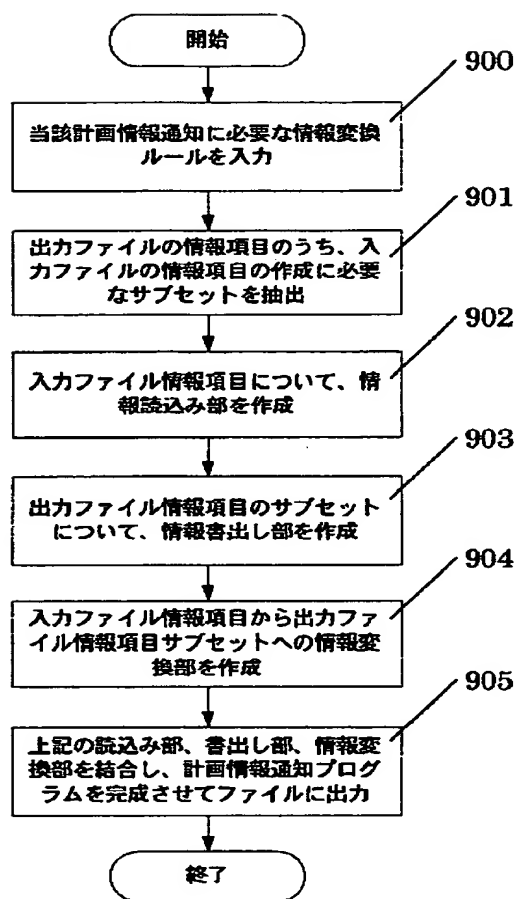
【図 8】

図 8



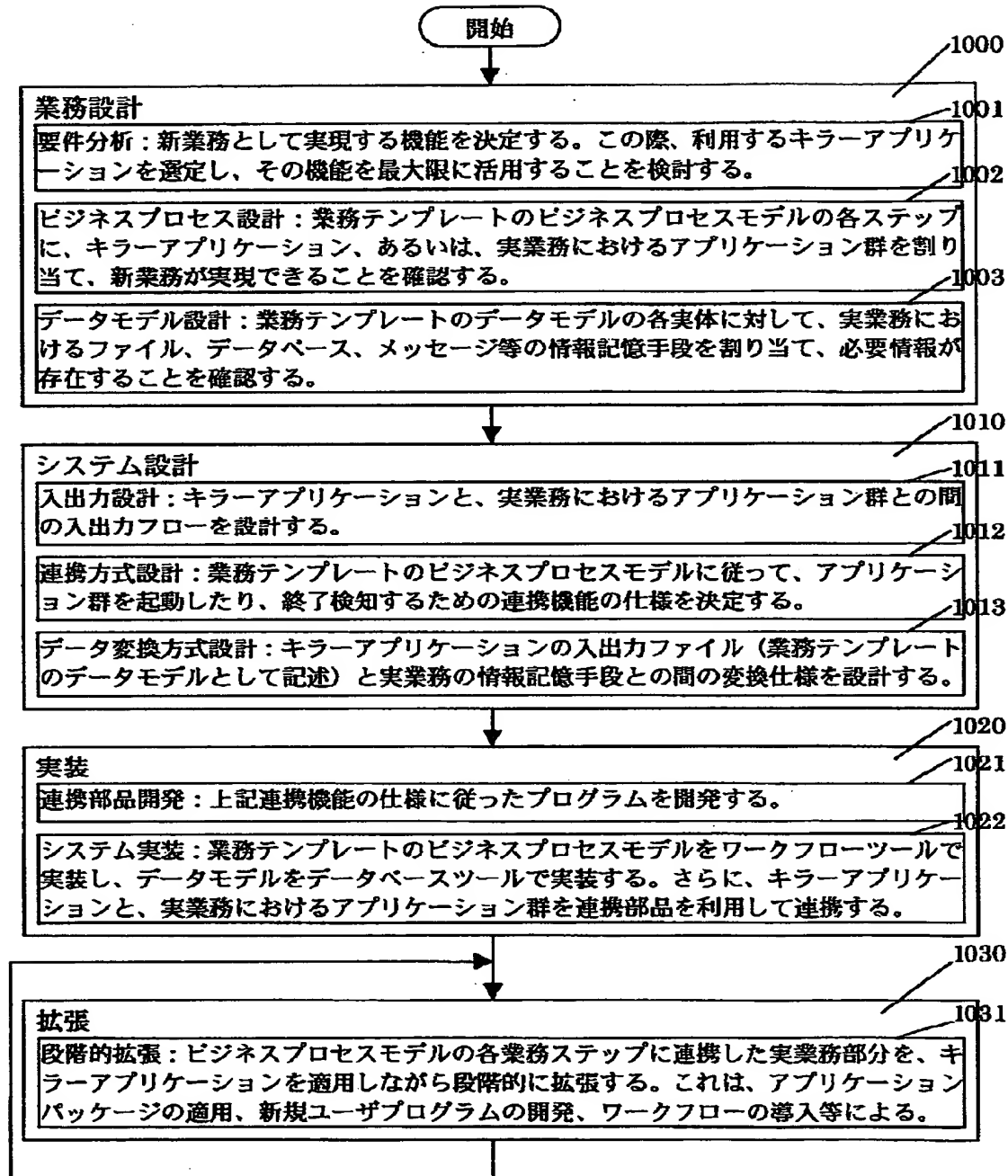
【図 9】

図 9



【図 10】

図 10



【書類名】 要約書

【要約】

【課題】

企業内の複数のアプリケーションを統合管理して新しい付加価値を生み出すキ
ラーアプリケーション適用において、従来の業務テンプレートによると実業務の
変更または業務テンプレートのカスタマイズが発生し、開発工数と期間が膨大に
なるという問題を解決する。

【解決手段】

本発明の業務テンプレートとして提供するビジネスプロセスモデルとデータモ
デルに従って、統合すべき業務間のマクロな手順を開発できる。さらに、業務連
携機能を用いて、現行ビジネスプロセスや既存アプリケーションをそのまま変更
することなく組込んで各業務内のミクロな手順を実現できる。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日	1990年 8月31日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台4丁目6番地
氏 名	株式会社日立製作所

INFORMATION UNDER 37 CFR 1.56(a)

(For Initial Filing)

The following references are submitted as information
to comply with the duty of disclosure under 37 CFR 1.56(a):

References	Disclosed in the specification?		Copy			Translation	
	Yes	No	Enc.	Follow	Please obtain	Enc.	Not available
1. JP-A-2000-172770		<input type="radio"/>	<input type="radio"/>			<input type="radio"/> (only abstract)	
2.							
3.							
4.							
5.							